

Doxxygen_MPDASTARONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 11:56:46 2006

Contents

1	Doxygen_MPDAUTHORITY Class Index	1
1.1	Doxygen_MPDAUTHORITY Class List	1
2	Doxygen_MPDAUTHORITY File Index	3
2.1	Doxygen_MPDAUTHORITY File List	3
3	Doxygen_MPDAUTHORITY Class Documentation	5
3.1	richScalarSender Class Reference	5
4	Doxygen_MPDAUTHORITY File Documentation	15
4.1	richScalarDaemon.cc File Reference	15
4.2	richScalarSender.cc File Reference	16
4.3	richScalarSender.hh File Reference	17
4.4	richScalarSender_i.cc File Reference	18
4.5	senderMain.cc File Reference	19

Chapter 1

Doxxygen_MPDU_STAR_ONLINE_DB Class Index

1.1 Doxygen_MPDU_STAR_ONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

richScalarSender	5
------------------	-------	---

Chapter 2

Doxxygen_MPDASTARONLINE_DB File Index

2.1 Doxygen_MPDASTARONLINE_DB File List

Here is a list of all files with brief descriptions:

richScalarDaemon.cc	15
richScalarSender.cc	16
richScalarSender.hh	17
richScalarSender_i.cc	18
senderMain.cc	19

Chapter 3

Doxxygen_MP_D_STAR_ONLINE_DB Class Documentation

3.1 richScalarSender Class Reference

```
#include <richScalarSender.hh>
```

Public Member Functions

- `richScalarSender` (const char *localDir)
dito
 - `virtual ~richScalarSender ()`
 - `virtual void initTable ()`
 - `virtual void initTags ()`
 - `virtual void initDataBase ()`
 - `virtual bool loadUserControls (const char *name, const char *value)`
 - `virtual void initQuery ()`
 - `virtual bool queryData ()`
 - `virtual bool readData (const char *fileName)`
 - `virtual bool updateDb (const char *fileName)`
 - `virtual bool readData (ifstream &from)`
 - `virtual bool hasChanged (int rowNumber)`
 - `char * readAny ()`
 - `bool readVal (char *&value)`
 - `bool readVal (float &value)`
 - `bool readVal (double &value)`
 - `bool readVal (short &value)`
 - `bool readVal (int &value)`
 - `bool readVal (long &value)`
 - `bool readVal (long long &value)`
 - `bool nextLine (ifstream &from)`
 - `void readError (int l, char *c, char *m)`

Protected Attributes

- richScalar `previousVals` [NUM_DB_ROWS]
- richScalar `tempVals` [NUM_DB_ROWS]
- int `elementList` [NUM_DB_ROWS]
- richScalar `updateVals` [NUM_DB_ROWS]
- int `updateElements` [NUM_DB_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tmpline` [256]
- char * `ptr1`
- char * `ptr2`

3.1.1 Constructor & Destructor Documentation

3.1.1.1 richScalarSender::richScalarSender (const char * *localDir*)

dito

Definition at line 18 of file richScalarSender.cc.

```

18
19
20     initTags();
21     if(localDir) cd(localDir); // note this ignores the sub dir tag
22     init("richScalar"); // setup the file I/O
23     initDataBase();      // database connections
24     initTable();         // table definitions
25
26 }
```

3.1.1.2 virtual richScalarSender::~richScalarSender () [inline, virtual]

Definition at line 41 of file richScalarSender.hh.

```
41 {};
```

3.1.2 Member Function Documentation

3.1.2.1 bool richScalarSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 107 of file richScalarSender_i.cc.

```

107
108
109 /***richScalar* pre=&previousVals[rowNumber];
110 richScalar* cur=&tempVals[rowNumber];
111
112 * example ... note -> change to any element requires db-update
113 * and thus returns true immediately
114 *
115 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
116 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
117 *
```

```

118  *  ....
119  */
120
121 return true;
122 }
```

3.1.2.2 void richScalarSender::initDataBase () [virtual]

Definition at line 74 of file richScalarSender.cc.

```

74
75 #define __METHOD__ "initDataBase( )"
76
77 /* More than an example... swap user & dbTrg as per subsystem*/
78 mgr->setUser("stardb","");
79 StDbType dbT = dbConditions;
80 StDbDomain dbD = dbRich;
81
82 if( !( node = mgr->initConfig(dbT,dbD)) )
83     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
84
85 #undef __METHOD__
86 }
```

3.1.2.3 void richScalarSender::initQuery () [virtual]

Definition at line 38 of file richScalarSender_i.cc.

```

38
39 #define __METHOD__ "initQuery()"
40
41     ofstream to(queryFile);
42
43     if(!to.is_open()){
44         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
45         return;
46     }
47     for(int i=1;i<17;i++)to<<"rich_1151E_val"<<i<<endl;
48 /* example
49 *      for(int i=0;i<16;i++){
50 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"<<endl;
51 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"<<endl;
52 *          ....
53 *
54 *      }
55 */
56
57     to.close();
58
59 #undef __METHOD__
60 }
```

3.1.2.4 void richScalarSender::initTable () [virtual]

Definition at line 29 of file richScalarSender.cc.

```

29
30 #define __METHOD__ "initTable()"
31
32     StDbTable* table=0;
33     if(!(table=node->addDbTable("richScalar")))
34         sendMess("Could not find table=richScalar",dbMFatal,__LINE__,__CLASS__,__METHOD__);
35
36     memset(tempVals,0,NUM_DB_ROWS*sizeof(richScalar));
37     memset(previousVals,0,NUM_DB_ROWS*sizeof(richScalar));
38
39     int nrows;
40     int* elist = table->getElementID(nrows);
41     if(nrows!=NUM_DB_ROWS){
42         //char mess[256];
43         ostringstream ms;
44         ms<<"Db rows("<<nrows<<") != compiled(";<<NUM_DB_ROWS;<<") "<<ends;
45         sendMess(ms.str().c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
46     }
47     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
48
49     unsigned int timestamp=time(NULL);
50     mgr->setRequestTime(timestamp);
51     if(mgr->fetchDbTable(table)){
52         richScalar* thv = (richScalar*)table->GetTable();
53         memcpy(previousVals,thv,nrows*sizeof(richScalar));
54     }
55
56 #undef __METHOD__
57 };

```

3.1.2.5 void richScalarSender::initTags () [virtual]

Definition at line 65 of file richScalarSender.cc.

```

65
66     /* more than an example -> swap "trg" to your subsys & add to email list*/
67     setEmailTo("porter@bnl.gov");
68     setDomainName("rich");
69
70 }

```

3.1.2.6 bool richScalarSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file richScalarSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25     if strstr(name,"driftLimit")){
26         driftLimit=atof(value);
27         sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28         return true;
29     }
30 */
31
32 return false;
33 #undef __METHOD__
34 }

```

3.1.2.7 bool richScalarSender::nextLine (ifstream &*from*) [inline]

Definition at line 74 of file richScalarSender.hh.

```
74
75     if(!from.getline(mline,255))return false;
76     return true;
77 }
```

3.1.2.8 bool richScalarSender::queryData () [virtual]

Definition at line 90 of file richScalarSender.cc.

```
90
91 #define __METHOD__ "queryData( )"
92
93 /*
94  * MORE THAN AN EXAMPLE....
95  * IF Standard SC-Query via "caGet" then,
96  * no need to change this method AT ALL
97  *
98 */
99
100    writeTime = (unsigned int)time(NULL);           //for database write time
101
102    //char systemCmd[1024];
103    ostringstream scmd;
104    scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
105
106    if(system((scmd.str()).c_str()))
107        return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109    return true;
110 #undef __METHOD__
111 };
```

3.1.2.9 char * richScalarSender::readAny ()

Definition at line 196 of file richScalarSender.cc.

```
196
197
198    strcpy(tmpline,mline);
199    ptr1=tmpline;
200    ptr2=strtok(ptr1," ");
201    if(!ptr2) return ptr2;
202    ptr2=strtok(NULL," ");
203    return ptr2;
204 }
```

3.1.2.10 bool richScalarSender::readData (ifstream &*from*) [virtual]

Definition at line 65 of file richScalarSender_i.cc.

```
65
66 #define __METHOD__ "readData(ifstream)"
```

```

67
68 mreadStatus=true;
69 memset(tempVals,0,NUM_DB_ROWS*sizeof(richScalar));
70
71
72 int i=0;
73 char* c=__CLASS__;
74 char* m=__METHOD__;
75
76 float data[16];
77
78 // this case we want to not write on any failure
79 // thus we return an error on reading failures
80 for(int i=0; i<16; i++)
81   if(!nextLine(from) || !readVal(data[i])) readError(__LINE__,c,m);
82
83 tempVals[0].rs1=data[0];
84   tempVals[0].rs2=data[1];
85   tempVals[0].rs3=data[2];
86   tempVals[0].rs4=data[3];
87   tempVals[0].rs5=data[4];
88   tempVals[0].rs6=data[5];
89   tempVals[0].rs7=data[6];
90   tempVals[0].rs8=data[7];
91   tempVals[0].rs9=data[8];
92   tempVals[0].rs10=data[9];
93   tempVals[0].rs11=data[10];
94   tempVals[0].rs12=data[11];
95   tempVals[0].rs13=data[12];
96   tempVals[0].rs14=data[13];
97   tempVals[0].rs15=data[14];
98   tempVals[0].rs16=data[15];
99
100 from.close();
101 return true;
102 #undef __METHOD__
103 }
```

3.1.2.11 bool richScalarSender::readData (const char *fileName) [virtual]

Definition at line 115 of file richScalarSender.cc.

```

115 {
116 #define __METHOD__ "readData(fileName)"
117
118   ifstream from(fileName);
119   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
120
121   return readData(from); // user implemented file read
122 #undef __METHOD__
123 }
```

3.1.2.12 void richScalarSender::readError (int l, char * c, char * m) [inline]

Definition at line 79 of file richScalarSender.hh.

```

79
80   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

3.1.2.13 bool richScalarSender::readVal (long long & value)

Definition at line 268 of file richScalarSender.cc.

```
268
269
270     if(!readAny())return false;
271     char* store[256];
272     value=strtoll(ptr2,store,10);
273     if(strlen(*store)>0) return false; // value is not a number
274
275     return true;
276 };
```

3.1.2.14 bool richScalarSender::readVal (long & value)

Definition at line 257 of file richScalarSender.cc.

```
257
258
259     if(!readAny())return false;
260
261     char* store[256];
262     value=strtol(ptr2,store,10);
263     if(strlen(*store)>0) return false; // value is not a number
264
265     return true;
266 };
```

3.1.2.15 bool richScalarSender::readVal (int & value)

Definition at line 246 of file richScalarSender.cc.

```
246
247
248     if(!readAny()) return false;
249
250     char* store[256];
251     value=(int)strtol(ptr2,store,10);
252     if(strlen(*store)>0) return false; // value is not a number
253
254     return true;
255 };
```

3.1.2.16 bool richScalarSender::readVal (short & value)

Definition at line 235 of file richScalarSender.cc.

```
235
236
237     if(!readAny()) return false;
238
239     char* store[256];
240     value=(short)strtol(ptr2,store,10);
241     if(strlen(*store)>0) return false; // value is not a number
242
243     return true;
244 };
```

3.1.2.17 bool richScalarSender::readVal (double & *value*)

Definition at line 224 of file richScalarSender.cc.

```
224
225
226     if(!readAny())return false;
227
228     char* store[256];
229     value=strtod(ptr2,store);
230     if(strlen(*store)>0) return false; // value is not a number
231
232     return true;
233 };
```

3.1.2.18 bool richScalarSender::readVal (float & *value*)

Definition at line 213 of file richScalarSender.cc.

```
213
214
215     if(!readAny()) return false;
216
217     char* store[256];
218     value=(float)strtod(ptr2,store);
219     if(strlen(*store)>0) return false; // value is not a number
220
221     return true;
222 };
```

3.1.2.19 bool richScalarSender::readVal (char *& *value*)

Definition at line 206 of file richScalarSender.cc.

```
206
207
208     if(!readAny()) return false;
209     strcpy(value,ptr2);
210
211 }
```

3.1.2.20 bool richScalarSender::updateDb (const char **fileName*) [virtual]

Definition at line 126 of file richScalarSender.cc.

```
126
127 #define __METHOD__ "updateDb(filename)"
128
129     if(!readData(fileName)) return sendMess(" Read data failed",dbMerr,__LINE__,__CLASS__,__METHOD__);
130
131     int* elements;
132     richScalar* vals;
133     int numRows = 0;
134
135     if(writeRequired()){

136 }
```

```

137     numRows=NUM_DB_ROWS;
138     elements=elementList;
139     vals = tempVals;
140
141 } else {
142
143     for(int i=0; i<NUM_DB_ROWS; i++){
144         if(hasChanged(i)){
145             updateElements[numRows]=elementList[i];
146             updateVals[numRows] = tempVals[i];
147             previousVals[i]=tempVals[i];
148             numRows++;
149         }
150     }
151
152     elements = updateElements;
153     vals      = updateVals;
154 }
155
156 if(numRows==0) return sendMess(" No update required for",mibName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
157
158 //char mess[256];
159 ostringstream sn;
160 sn<<"Will Update "<<numRows<< " of "<<NUM_DB_ROWS<<" rows "<<ends;
161 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
162
163 StDbTable* dbTable=node->findTable("richScalar");
164 dbTable->SetTable((char*)vals, numRows, elements);
165 mgr->setStoreTime(writeTime);
166
167 if(!mgr->storeDbTable(dbTable)) {
168     addBackLog(writeTime);
169     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
170 }
171
172 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
173
174 return true;
175 #undef __METHOD__
176 }
```

3.1.3 Member Data Documentation

3.1.3.1 int richScalarSender::elementList[NUM_DB_ROWS] [protected]

Definition at line 24 of file richScalarSender.hh.

3.1.3.2 char richScalarSender::mline[256] [protected]

Definition at line 29 of file richScalarSender.hh.

3.1.3.3 bool richScalarSender::mreadStatus [protected]

Definition at line 28 of file richScalarSender.hh.

3.1.3.4 richScalar richScalarSender::previousVals[NUM_DB_ROWS] [protected]

Definition at line 22 of file richScalarSender.hh.

3.1.3.5 char* richScalarSender::ptr1 [protected]

Definition at line 31 of file richScalarSender.hh.

3.1.3.6 char * richScalarSender::ptr2 [protected]

Definition at line 31 of file richScalarSender.hh.

3.1.3.7 richScalar richScalarSender::tempVals[NUM_DB_ROWS] [protected]

Definition at line 23 of file richScalarSender.hh.

3.1.3.8 char richScalarSender::tmpline[256] [protected]

Definition at line 30 of file richScalarSender.hh.

3.1.3.9 int richScalarSender::updateElements[NUM_DB_ROWS] [protected]

Definition at line 26 of file richScalarSender.hh.

3.1.3.10 richScalar richScalarSender::updateVals[NUM_DB_ROWS] [protected]

Definition at line 25 of file richScalarSender.hh.

The documentation for this class was generated from the following files:

- [richScalarSender.hh](#)
- [richScalarSender.cc](#)
- [richScalarSender_i.cc](#)

Chapter 4

Doxxygen_MPDAEMON File Documentation

4.1 richScalarDaemon.cc File Reference

```
#include "richScalarSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char **ldir*)

4.1.1 Function Documentation

4.1.1.1 void [runSender](#) (const char * *ldir*)

Definition at line 14 of file richScalarDaemon.cc.

```
14 {  
15  
16     CndDbSender* sender = new richScalarSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24 };
```

4.2 richScalarSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "richScalarSender.hh"
#include "StDbTable.h"
#include "richScalarSender_i.cc"
```

Defines

- #define **_CLASS_** "richScalarSender"
- #define **_METHOD_** "initTable()"
- #define **_METHOD_** "initDataBase()"
- #define **_METHOD_** "queryData()"
- #define **_METHOD_** "readData(fileName)"
- #define **_METHOD_** "updateDb(filename)"

4.2.1 Define Documentation

4.2.1.1 #define **_CLASS_** "richScalarSender"

Definition at line 16 of file richScalarSender.cc.

4.2.1.2 #define **_METHOD_** "updateDb(filename)"

4.2.1.3 #define **_METHOD_** "readData(fileName)"

4.2.1.4 #define **_METHOD_** "queryData()"

4.2.1.5 #define **_METHOD_** "initDataBase()"

4.2.1.6 #define **_METHOD_** "initTable()"

4.3 richScalarSender.hh File Reference

```
#include "CndDbSender.hh"
#include "richScalar.h"
```

Classes

- class [richScalarSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.3.1 Define Documentation

4.3.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file richScalarSender.hh.

4.4 richScalarSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.4.1 Define Documentation

4.4.1.1 #define METHOD "readData(ifstream)"

4.4.1.2 #define METHOD "initQuery()"

4.4.1.3 #define METHOD "loadUserControls(name,value)"

4.5 senderMain.cc File Reference

```
#include <unistd.h>
#include <sys/stat.h>
#include <iostream.h>
```

Functions

- void **runSender** (const char *directory)
- int **main** (int argc, char *argv[])

4.5.1 Function Documentation

4.5.1.1 int main (int *argc*, char * *argv*[])

Definition at line 16 of file senderMain.cc.

```
16
17
18
19     /*pid_t pid;
20     if((pid = fork()) < 0) cout<<"Unable to fork process";
21     else if(pid != 0) exit(0);
22     setsid();*/
23 //run as session leader
24     chdir("/");
25     umask(0); //change umask
26
27     char* localDirectory = 0;
28
29     for (int i=1; i<argc;i++){
30         if(argv[i][0] == '-'){
31             switch (argv[i][1]) {
32                 case 'c' :
33                     localDirectory = argv[++i];
34                     break;
35                 default  :
36                     break;
37             }
38         }
39     }
40
41     runSender(localDirectory);
42
43 return 0;
44 };
```

4.5.1.2 void runSender (const char * *directory*)

Definition at line 14 of file richScalarDaemon.cc.

```
14
15
16     CndDbSender* sender = new richScalarSender(ldir);
17
18     sender->initQuery();
```

```
19   for(;;) { //ever...
20     if(sender->hasBackLog())sender->cleanBackLog();
21     if(sender->queryData())sender->updateDb();
22     sleep(sender->sleepTime());
23   }
24
25 };
```

Index

~richScalarSender
 richScalarSender, 6
__CLASS__
 richScalarSender.cc, 16
__METHOD__
 richScalarSender.cc, 16
 richScalarSender_i.cc, 18

elementList
 richScalarSender, 13

hasChanged
 richScalarSender, 6

initDataBase
 richScalarSender, 7
initQuery
 richScalarSender, 7
initTable
 richScalarSender, 7
initTags
 richScalarSender, 8

loadUserControls
 richScalarSender, 8

main
 senderMain.cc, 19
mline
 richScalarSender, 13
mreadStatus
 richScalarSender, 13

nextLine
 richScalarSender, 8
NUM_DB_ROWS
 richScalarSender.hh, 17

previousVals
 richScalarSender, 13
ptr1
 richScalarSender, 13
ptr2
 richScalarSender, 14

queryData
 richScalarSender, 9
richScalarSender, 9

readAny
 richScalarSender, 9
readData
 richScalarSender, 9, 10
readError
 richScalarSender, 10
readVal
 richScalarSender, 10–12
richScalarDaemon.cc, 15
richScalarDaemon.cc
 runSender, 15
richScalarSender, 5
 richScalarSender, 6
richScalarSender
 ~richScalarSender, 6
 elementList, 13
 hasChanged, 6
 initDataBase, 7
 initQuery, 7
 initTable, 7
 initTags, 8
 loadUserControls, 8
 mline, 13
 mreadStatus, 13
 nextLine, 8
 previousVals, 13
 ptr1, 13
 ptr2, 14
 queryData, 9
 readAny, 9
 readData, 9, 10
 readError, 10
 readVal, 10–12
 richScalarSender, 6
 tempVals, 14
 tmpline, 14
 updateDb, 12
 updateElements, 14
 updateVals, 14
richScalarSender.cc, 16
richScalarSender.cc
 __CLASS__, 16
 __METHOD__, 16

richScalarSender.hh, 17
richScalarSender.hh
 NUM_DB_ROWS, 17
richScalarSender_i.cc, 18
richScalarSender_i.cc
 __METHOD__, 18
runSender
 richScalarDaemon.cc, 15
 senderMain.cc, 19

senderMain.cc, 19
senderMain.cc
 main, 19
 runSender, 19

tempVals
 richScalarSender, 14
tmpline
 richScalarSender, 14

updateDb
 richScalarSender, 12
updateElements
 richScalarSender, 14
updateVals
 richScalarSender, 14